

编译

2020年7月6日 23:10

1. 编译

- a. 机器语言 (Machine Language): 可以被计算机直接理解
 - i. 与人类表达习惯相去甚远
 - ii. 难记忆
 - iii. 难编写、难阅读
 - iv. 易写错
- b. 汇编语言 (Assembly Language): 引入助记符
 - i. 依赖于特定机器, 非计算机专业人员使用受限制
 - ii. 编写效率依然很低
- c. 高级语言 (High Level Language): 类似于数学定义或自然语言的简洁形式
 - i. 接近人类表达习惯
 - ii. 不依赖于特定机器
 - iii. 编写效率高
- d. 编译: 将高级语言翻译成汇编语言或机器语言的过程
 - i. 即直接翻译成及其语言, 或仅翻译到汇编语言的过程都是编译
 - ii. 汇编语言翻译成机器语言的过程叫汇编
- e. 语言处理系统
 - i. 源程序>预处理器(Preprocessor)>经过预处理的源程序
 - 1) 把存储在不同文件中的源程序聚合在一起
 - 2) 把被称为宏的缩写语句转换为原始语句
 - ii. 经过预处理的源程序>编译器>汇编语言程序
 - iii. 汇编语言程序>汇编器(Assembler)>可重定位的机器代码
 - 1) 可重定位(Relocatable): 在内存中存放的起始位置L不是固定的
 - iv. 可重定位的机器代码>链接器(Linker)或加载器(Loader)>目标机器代码
 - 1) 加载器: 修改可重定位地址; 将修改后的指令和数据放到内存中适当的位置
 - 2) 起始位置+相对地址=绝对地址
 - 3) 链接器: 将多个可重定位的机器代码文件(包括库文件)连接到一起, 解决外部内存地址问题

2. 编译系统结构

- a. 与源语言相关的分析部分称为前端front-end
 - i. 词法分析 (Lexical Analysis) 从字符流得到词法单元流
 - ii. 语法分析 (Syntax Analysis) 从词法单元流得到语法树
 - iii. 语义分析 (Semantic Analysis) 扩展语法树
- b. 可选的中间代码生成器和机器无关代码优化器
- c. 与目标语言相关的综合部分称为后端back-end

- i. 目标代码生成器从中间表示形式生成目标机器语言
 - ii. 及其相关代码优化器生成目标机器语言
- 3. 词法分析/扫描(Scanning)
 - a. 主要任务：从左向右逐行扫描源程序的字符，识别出各个单词，确定单词的类型。将识别出的单词转换成统一的机内表示——词法单元(token)形式。形如
token: < 种别码, 属性值 >
- 4. 语法分析(parsing)
 - a. 语法分析器(parser)从词法分析器输出的token序列中 识别出各类短语，并构造语法分析树(parse tree)。用于描述句子的语法结构
- 5. 语义分析
 - a. 收集标识符的属性信息，包括种属(Kind)，类型(Type)，存储位置、长度，值，作用域，参数、返回值
 - b. 语义检查：
 - i. 变量或过程未经声明就使用
 - ii. 变量或过程名重复声明
 - iii. 运算分量类型不匹配
 - iv. 操作符与操作数之间的类型不匹配
 - v. 数组下标不是整数
 - vi. 对非数组变量使用数组访问操作符
 - vii. 对非过程名使用过程调用操作符
 - viii. 过程调用的参数类型或数目不匹配
 - ix. 函数返回类型有误
- 6. 中间代码生成及编译器后端
 - a. 常用中间表示形式：
 - i. 三地址码(Three-address Code)：由类似于汇编语言的指令序列组成，每个指令最多有三个操作数(operand)
 - 1) 地址可以具有如下形式之一
 - a) 源程序中的名字(name)
 - b) 常量(constant)
 - c) 编译器生成的临时变量(temporary)
 - 2) 表示形式
 - a) 四元式(Quadruples)形如(op, y, z, x)
 - b) 三元式(Triples)
 - c) 间接三元式(Indirect triples)
 - ii. 语法结构树/语法树(Syntax Trees)
 - b. 作用
 - i. 目标代码生成以源程序的中间表示形式作为输入，并把它映射到目标语言
 - ii. 目标代码生成的一个重要任务是为程序中使用的变量合理分配寄存器
 - iii. 代码优化：为改进代码所进行的等价程序变换，使其运行得更快一些、占用空间更少一些，或者二者兼顾